DevOops

SERVERLESS

Проблемы разработки

ПЛАН

1. Мир Serverless
2. Особенности реализации
3. Выводы

О СЕБЕ

Руслан Серкин

Software engineer
DataArt

Email: ruslan.ru@gmail.com
Twitter: @ruslan_firelfy

ПРОЕКТ

- Автоматизация оффлайн бизнеса

ПРОЕКТ

- Автоматизация оффлайн бизнеса
- Команда: JavaScript разработчики

ПРОЕКТ

- Автоматизация оффлайн бизнеса
- Команда: JavaScript разработчики
- AWS

SERVERLESS

SERVERLESS - ПРИМЕНЕНИЕ

- Sites / API

SERVERLESS - ПРИМЕНЕНИЕ

- Sites / API
- Internet of things

SERVERLESS - ПРИМЕНЕНИЕ

- Sites / API
- Internet of things
- Переодические задания

SERVERLESS - ПРИМЕНЕНИЕ

- Sites / API
- Internet of things
- Переодические задания
- Рассылка писем

SERVERLESS - ПРИМЕНЕНИЕ

- Sites / API
- Internet of things
- Переодические задания
- Рассылка писем
- Сервисные задания

SERVERLESS - ПРИМЕНЕНИЕ

- Sites / API
- Internet of things
- Переодические задания
- Рассылка писем
- Сервисные задания
- Другое

SERVERLESS - ПРОВАЙДЕРЫ

- AWS Lambda (AWS)

SERVERLESS - ПРОВАЙДЕРЫ

- AWS Lambda (AWS)
- Azure Functions (Azure)

SERVERLESS - ПРОВАЙДЕРЫ

- AWS Lambda (AWS)
- Azure Functions (Azure)
- Cloud Functions (GCP)

SERVERLESS - ПРОВАЙДЕРЫ

- AWS Lambda (AWS)
- Azure Functions (Azure)
- Cloud Functions (GCP)
- Kubeless (K8s)

SERVERLESS - ПРОВАЙДЕРЫ

- AWS Lambda (AWS)
- Azure Functions (Azure)
- Cloud Functions (GCP)
- Kubeless (K8s)
- Другие решения

SERVERLESS - ПРЕИМУЩЕСТВА

- Просто функция

SERVERLESS - ПРИМЕР (AWS)

```
"use strict";
module.exports.endpoint = (event, context, callback) ⇒ {
  const response = {
    statusCode: 200,
    body: JSON.stringify({
      message: `Hello, the current time is
                              ${new Date().toTimeString()}.`
    })
  };

  callback(null, response);
};
```

SERVERLESS - ПРЕИМУЩЕСТВА

- Просто функция
  - Легко тестировать

SERVERLESS - ПРЕИМУЩЕСТВА

- Просто функция
  - Легко тестировать
  - Меньше зависимостей

SERVERLESS - ПРЕИМУЩЕСТВА

- Просто функция
  - Легко тестировать
  - Меньше зависимостей
  - Идеальный микросервис

SERVERLESS - ПРЕИМУЩЕСТВА

- Деплой

SERVERLESS - ПРЕИМУЩЕСТВА

- Деплой
  - Zip архив

SERVERLESS - ПРЕИМУЩЕСТВА

- Деплой
  - Zip архив
  - Интерфейс

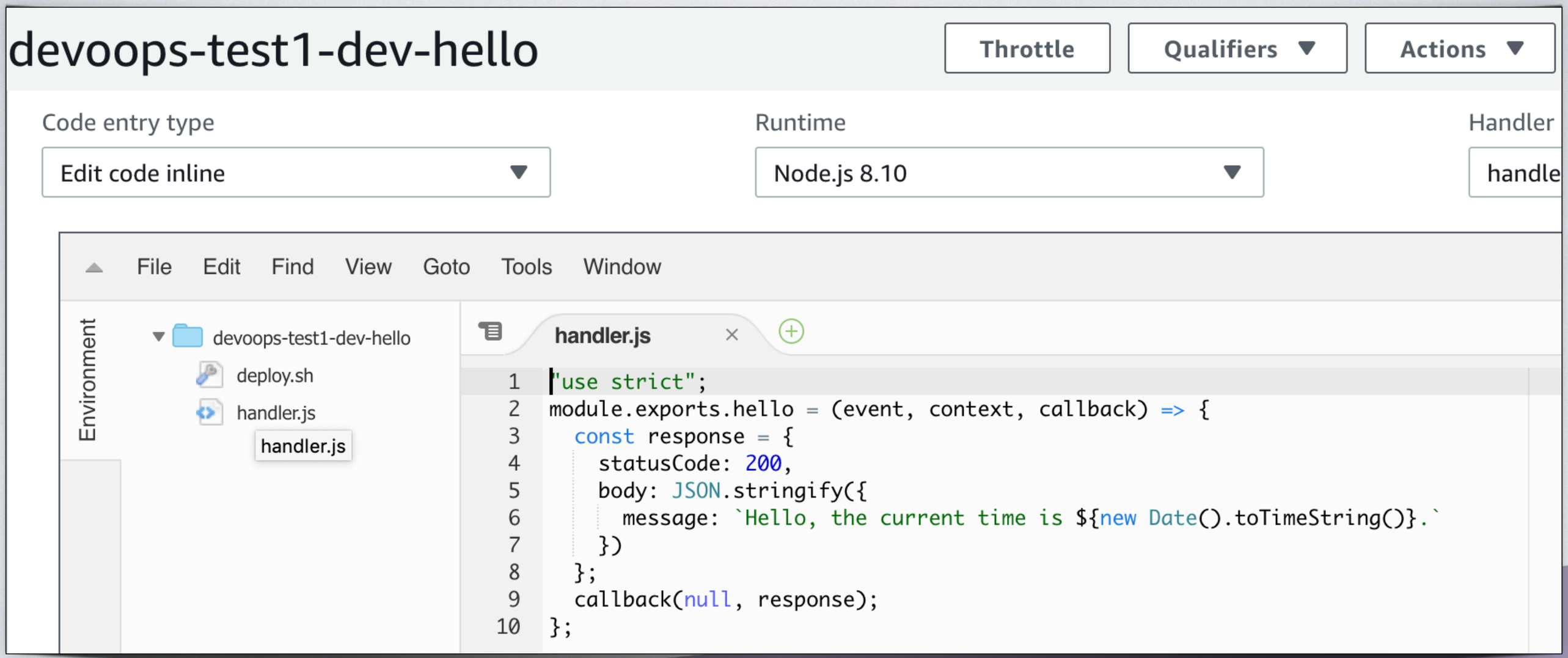
SERVERLESS - ПРЕИМУЩЕСТВА

- Деплой
  - Zip архив
  - Интерфейс
  - Google repos

SERVERLESS - ПРЕИМУЩЕСТВА

- Деплой
  - Zip архив
  - Интерфейс
  - Google repos
  - Утилиты и плагины для IDE

SERVERLESS - AWS CLOUD9



SERVERLESS - ПРЕИМУЩЕСТВА

- Масштабируемость

SERVERLESS - ПРЕИМУЩЕСТВА

- Масштабируемость
  - Нет серверов

SERVERLESS - ПРЕИМУЩЕСТВА

- Масштабируемость
  - Нет серверов
  - Функция на событие (API Gateway, SNS, SQS…)

SERVERLESS - ПРЕИМУЩЕСТВА

- Масштабируемость
  - Нет серверов
  - Функция на событие (API Gateway, SNS, SQS…)
  - Быстрый старт

SERVERLESS - ПРЕИМУЩЕСТВА

- Цена

SERVERLESS - ПРЕИМУЩЕСТВА

- Цена
  - Нет месячной подписки

SERVERLESS - ПРЕИМУЩЕСТВА

- Цена
  - Нет месячной подписки
  - Платим за время и ресурсы

SERVERLESS - ПРЕИМУЩЕСТВА

- Цена
  - Нет месячной подписки
  - Платим за время и ресурсы
  - serverlesscalc.com

SERVERLESS - ПРЕИМУЩЕСТВА

- Языки программирования

SERVERLESS - ПРЕИМУЩЕСТВА

- Языки программирования
  - Python

SERVERLESS - ПРЕИМУЩЕСТВА

- Языки программирования
  - Python
  - NodeJS

SERVERLESS - ПРЕИМУЩЕСТВА

- Языки программирования
  - Python
  - NodeJS
  - Go

SERVERLESS - ПРЕИМУЩЕСТВА

- Языки программирования
  - Python
  - NodeJS
  - Go
  - Java

SERVERLESS - ПРЕИМУЩЕСТВА

- Языки программирования
  - Python
  - NodeJS
  - Go
  - Java
  - C#

СРАВНЕНИЕ ЯЗЫКОВ НА AWS

	Типы				
GO	✅✅				
NodeJS	❌				
Python	❌				
Java	✅✅				
C#	✅✅				

СРАВНЕНИЕ ЯЗЫКОВ НА AWS

	Типы	Деплой			
GO	✅✅	✅			
NodeJS	❌	✅✅			
Python	❌	✅✅			
Java	✅✅	❌			
C#	✅✅	✅			

СРАВНЕНИЕ ЯЗЫКОВ НА AWS

	Типы	Деплой	Старт		
GO	✅✅	✅	✅		
NodeJS	❌	✅✅	✅✅		
Python	❌	✅✅	✅✅		
Java	✅✅	❌	❌		
C#	✅✅	✅	❌		

СРАВНЕНИЕ ЯЗЫКОВ НА AWS

	Типы	Деплой	Старт	Скорость	
GO	✅✅	✅	✅	✅✅	
NodeJS	❌	✅✅	✅✅	✅	
Python	❌	✅✅	✅✅	✅	
Java	✅✅	❌	❌	✅✅	
C#	✅✅	✅	❌	✅✅	

СРАВНЕНИЕ ЯЗЫКОВ НА AWS

	Типы	Деплой	Старт	Скорость	Экосистема
GO	✅✅	✅	✅	✅✅	✅✅
NodeJS	❌	✅✅	✅✅	✅	✅
Python	❌	✅✅	✅✅	✅	✅✅
Java	✅✅	❌	❌	✅✅	❌
C#	✅✅	✅	❌	✅✅	❌

СРАВНЕНИЕ ПЛАТФОРМ

	AWS	Azure	GCP	Kubeless
Деплой	ZIP / Web Editor	ZIP / Web Editor	ZIP/ Google repo	CLI

СРАВНЕНИЕ ПЛАТФОРМ

	AWS	Azure	GCP	Kubeless
Деплой	ZIP / Web Editor	ZIP / Web Editor	ZIP/ Google repo	CLI
Языки	Java, JS, C#, Go, Python	C#, F#, JS, Python, Java, PHP	JavaScript	Python, JS, Go, Ruby, PHP, C#

СРАВНЕНИЕ ПЛАТФОРМ

	AWS	Azure	GCP	Kubeless
Деплой	ZIP / Web Editor	ZIP / Web Editor	ZIP/ Google repo	CLI
Языки	Java, JS, C#, Go, Python	C#, F#, JS, Python, Java, PHP	JavaScript	Python, JS, Go, Ruby, PHP, C#
Зависимости	стандартные пакеты языка	стандартные пакеты языка	стандартные пакеты языка	стандартные пакеты языка

СРАВНЕНИЕ ПЛАТФОРМ

	AWS	Azure	GCP	Kubeless
Деплой	ZIP / Web Editor	ZIP / Web Editor	ZIP/ Google repo	CLI
Языки	Java, JS, C#, Go, Python	C#, F#, JS, Python, Java, PHP	JavaScript	Python, JS, Go, Ruby, PHP, C#
Зависимости	стандартные пакеты языка	стандартные пакеты языка	стандартные пакеты языка	стандартные пакеты языка
Триггеры	AWS Services	AZURE Services	Cloud pub/sub, Storage, HTTP	Kafka, NATS, HTTP

СРАВНЕНИЕ ПЛАТФОРМ

	AWS	Azure	GCP	Kubeless
Деплой	ZIP / Web Editor	ZIP / Web Editor	ZIP/ Google repo	CLI
Языки	Java, JS, C#, Go, Python	C#, F#, JS, Python, Java, PHP	JavaScript	Python, JS, Go, Ruby, PHP, C#
Зависимости	стандартные пакеты языка	стандартные пакеты языка	стандартные пакеты языка	стандартные пакеты языка
Триггеры	AWS Services	AZURE Services	Cloud pub/sub, Storage, HTTP	Kafka, NATS, HTTP
Мониторинг	CloudWatch, X-Ray	Application Insights	StackDriver	Prometheus

ПРОЕКТ

- Мобильные приложения

ПРОЕКТ

- Мобильные приложения
- Сайт

ПРОЕКТ

- Мобильные приложения
- Сайт
- Точки продаж с системой распознавания QR

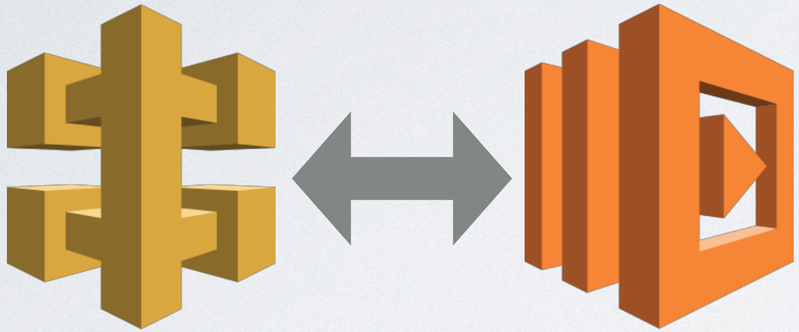
ПРОЕКТ

- Мобильные приложения
- Сайт
- Точки продаж с системой распознавания QR
- Несколько регионов

КОНФИГУРАЦИЯ

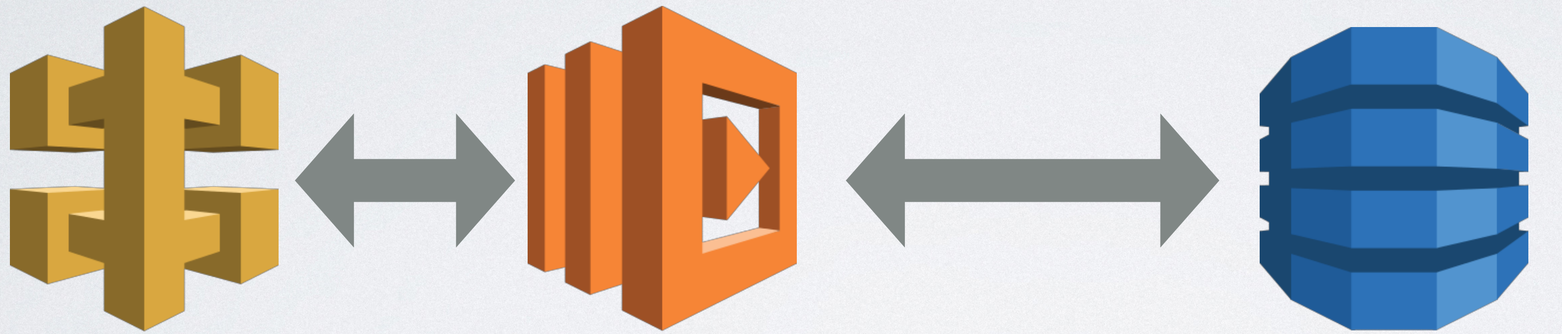
API GateWay

КОНФИГУРАЦИЯ



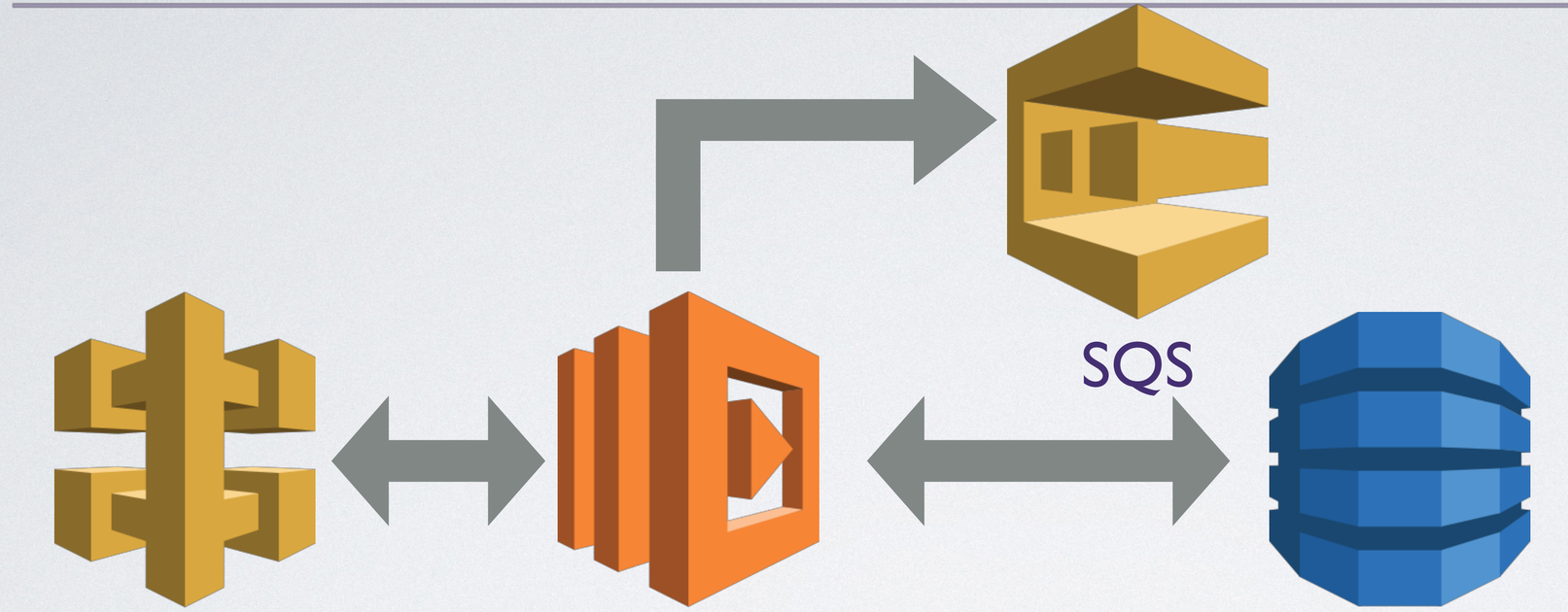
API GateWay Lambda

КОНФИГУРАЦИЯ



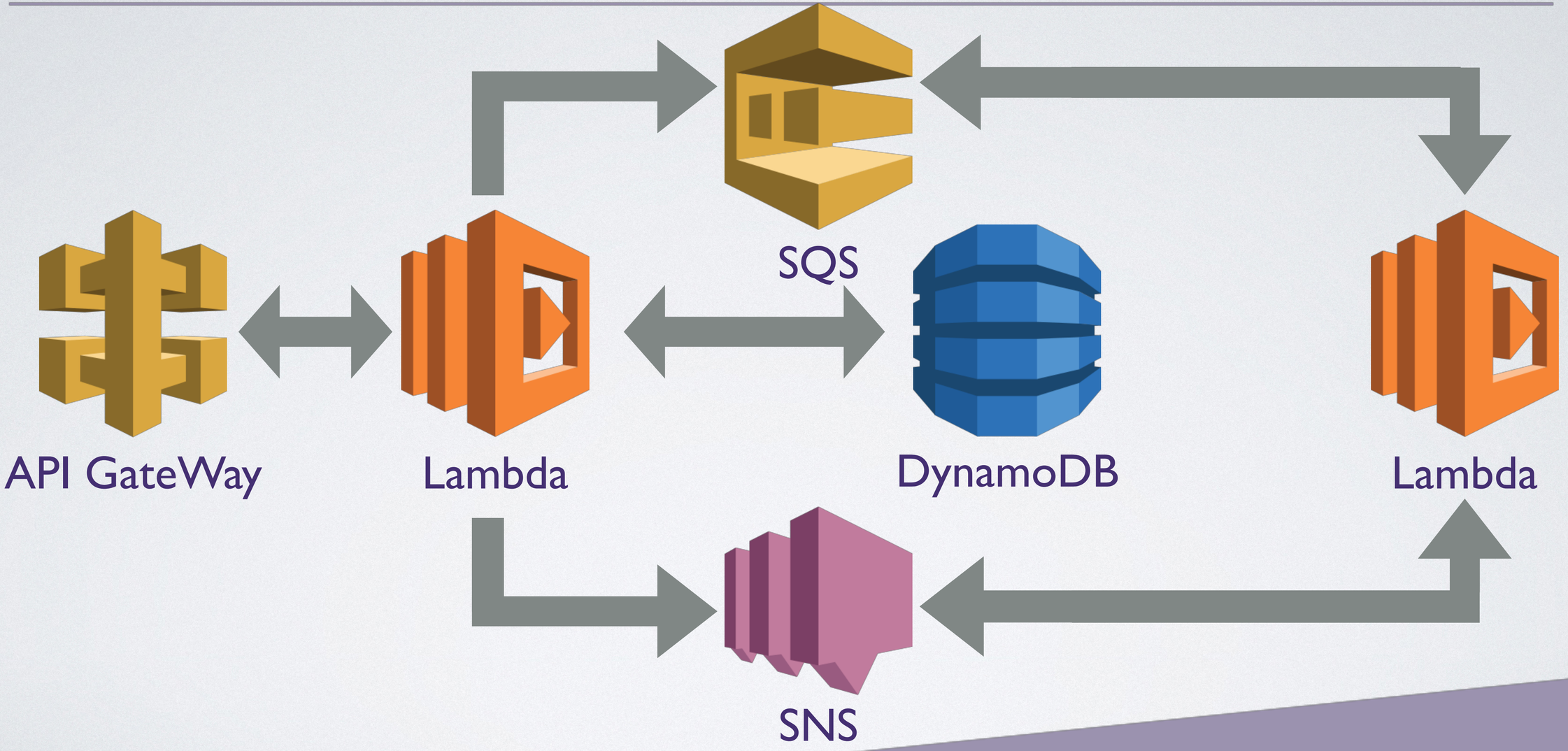
API GateWay Lambda DynamoDB

КОНФИГУРАЦИЯ

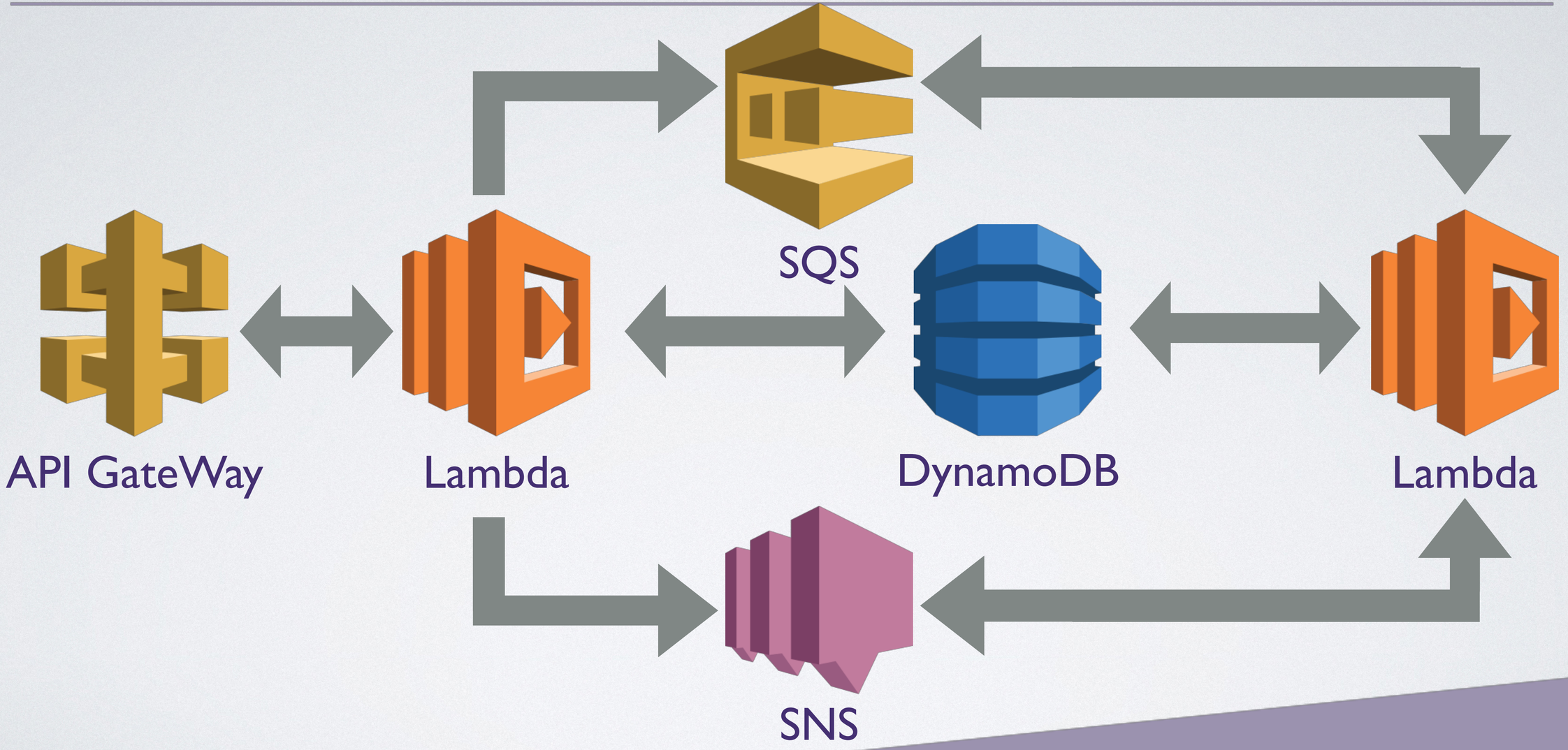


КОНФИГУРАЦИЯ

КОНФИГУРАЦИЯ



КОНФИГУРАЦИЯ



ЛОГИРОВАНИЕ

ЛОГИРОВАНИЕ

- AWS CloudWatch

ЛОГИРОВАНИЕ

- AWS CloudWatch
- ELK

ЛОГИРОВАНИЕ

- AWS CloudWatch
- ELK
- Внешний сервис

ЛОГИРОВАНИЕ - CLOUDWATCH

- Простота интеграции

ЛОГИРОВАНИЕ - CLOUDWATCH

- Простота интеграции
- Готовая инфраструктура

ЛОГИРОВАНИЕ - CLOUDWATCH

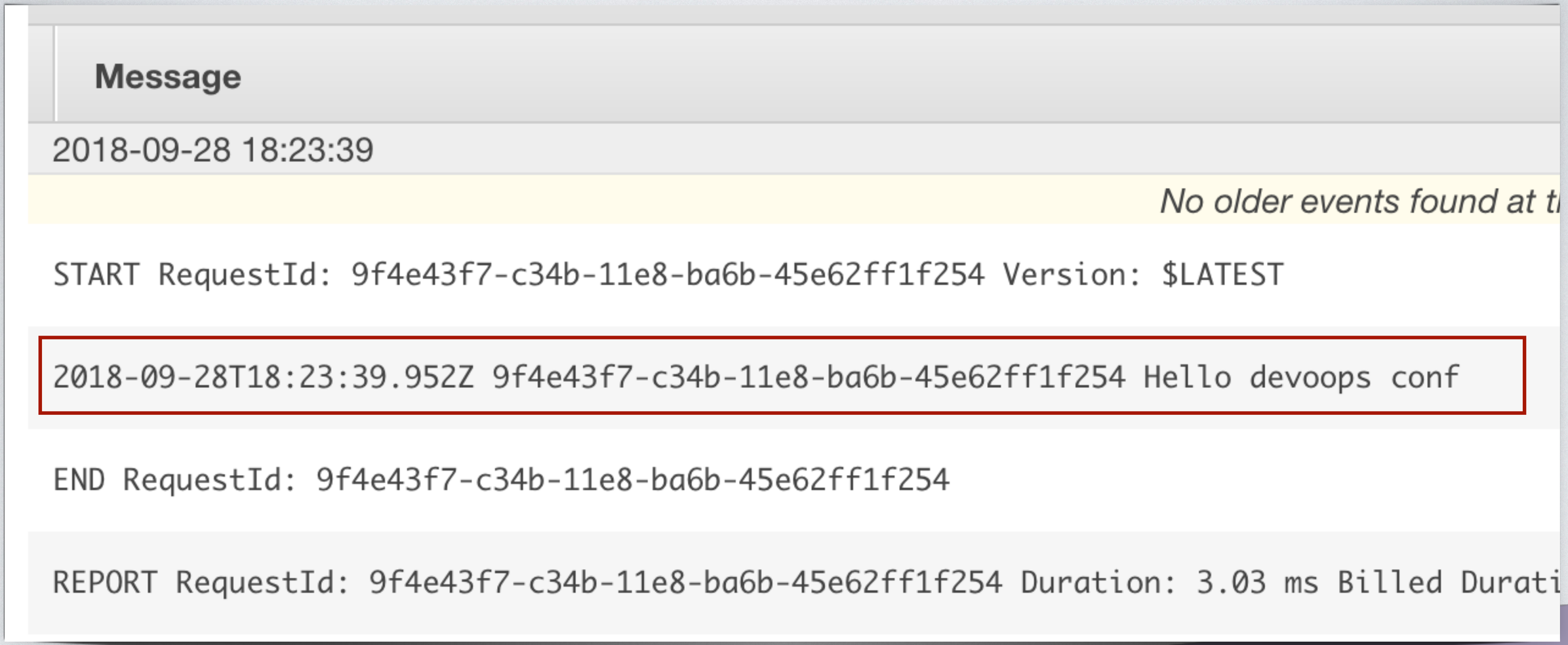
- Простота интеграции
- Готовая инфраструктура
- Пишем в stdout

ЛОГИРОВАНИЕ - CLOUDWATCH

handler.js

```
1  "use strict";
2  module.exports.hello = (event, context, callback) => {
3
4    console.log("Hello devoops conf");
5
6    const response = {
7      statusCode: 200,
8      body: JSON.stringify({
9        message: `Hello, the current time is ${new Date().toTimeString()}.`
10     })
11   };
12   callback(null, response);
13 };
```

ЛОГИРОВАНИЕ - CLOUDWATCH



ЛОГИРОВАНИЕ - CLOUDWATCH

- Неудобно читать логи

ЛОГИРОВАНИЕ - CLOUDWATCH

- Неудобно читать логи
- Тяжело построить контекст

ЛОГИРОВАНИЕ - CLOUDWATCH

- Неудобно читать логи
- Тяжело построить контекст
- X-Ray (http://clc.to/JQi-zA)

ЛОГИРОВАНИЕ - CLOUDWATCH

- Неудобно читать логи
- Тяжело построить контекст
- X-Ray (http://clc.to/JQi-zA)
  - нет данных в реальном времени

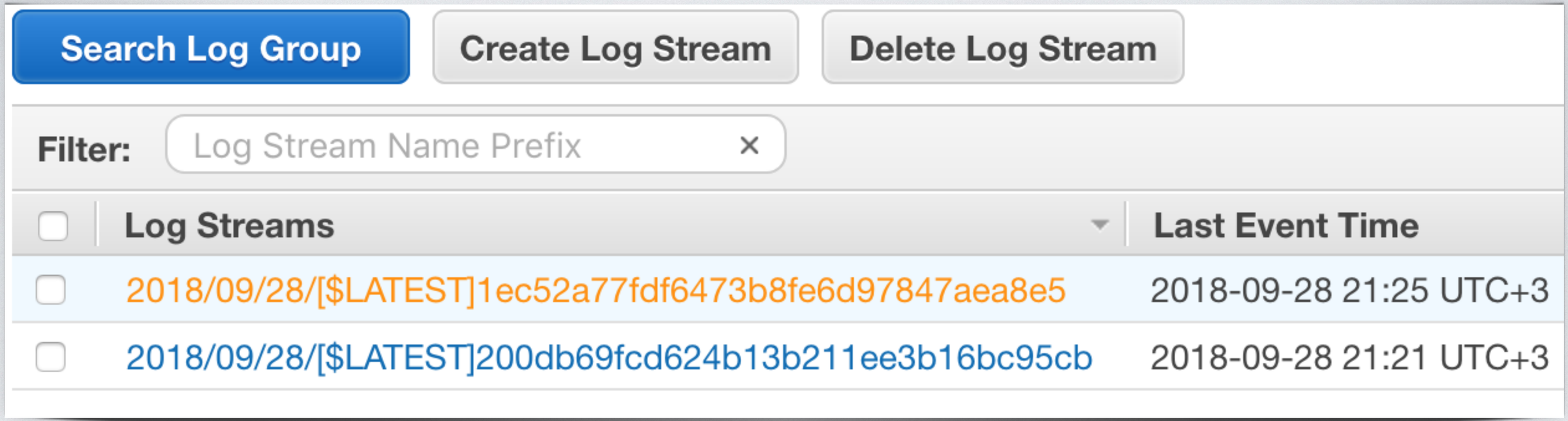
ЛОГИРОВАНИЕ - CLOUDWATCH

- Неудобно читать логи
- Тяжело построить контекст
- X-Ray (http://clc.to/JQi-zA)
  - нет данных в реальном времени
  - не всегда данные достоверны

ЛОГИРОВАНИЕ - CLOUDWATCH

- Неудобно читать логи
- Тяжело построить контекст
- X-Ray (http://clc.to/JQi-zA)
  - нет данных в реальном времени
  - не всегда данные достоверны
- Логи сгруппированы

ЛОГИРОВАНИЕ - CLOUDWATCH



ЛОГИРОВАНИЕ - ELK / EXTERNAL

- Куда писать

ЛОГИРОВАНИЕ - ELK / EXTERNAL

- Куда писать
  - AWS CloudWatch

ЛОГИРОВАНИЕ - ELK / EXTERNAL

- Куда писать
  - AWS CloudWatch
  - Внешний сервис из кода

ЛОГИРОВАНИЕ - ELK / EXTERNAL

- Куда писать
  - AWS CloudWatch
  - Внешний сервис из кода
  - Stdout

ЛОГИРОВАНИЕ - ELK / EXTERNAL

- Куда писать
  - AWS CloudWatch
  - Внешний сервис из кода
  - Stdout
- Простой поиск

ЛОГИРОВАНИЕ - ELK / EXTERNAL

- Куда писать
  - AWS CloudWatch
  - Внешний сервис из кода
  - Stdout
- Простой поиск
- Построение графиков

ЛОГИРОВАНИЕ - TIMEOUTS

- Функция не работает бесконечно

ЛОГИРОВАНИЕ - TIMEOUTS

- Функция не работает бесконечно
  - Максимально 300 секунд

ЛОГИРОВАНИЕ - TIMEOUTS

- Функция не работает бесконечно
  - Максимально 300 секунд
- Потеря данных

ЛОГИРОВАНИЕ - TIMEOUTS

- Функция не работает бесконечно
  - Максимально 300 секунд
- Потеря данных
  - Повторный запуск после ошибки

ЛОГИРОВАНИЕ - TIMEOUTS

- Функция не работает бесконечно
  - Максимально 300 секунд
- Потеря данных
  - Повторный запуск после ошибки
  - Создавать свой таймер

ДЕПЛОЙ
И
КОНФИГУРАЦИЯ

КОНФИГУРАЦИЯ И ДЕПЛОЙ

- Интерфейс

КОНФИГУРАЦИЯ И ДЕПЛОЙ

- Интерфейс
- CLI

КОНФИГУРАЦИЯ И ДЕПЛОЙ

- Интерфейс
- CLI
- Плагины для IDE

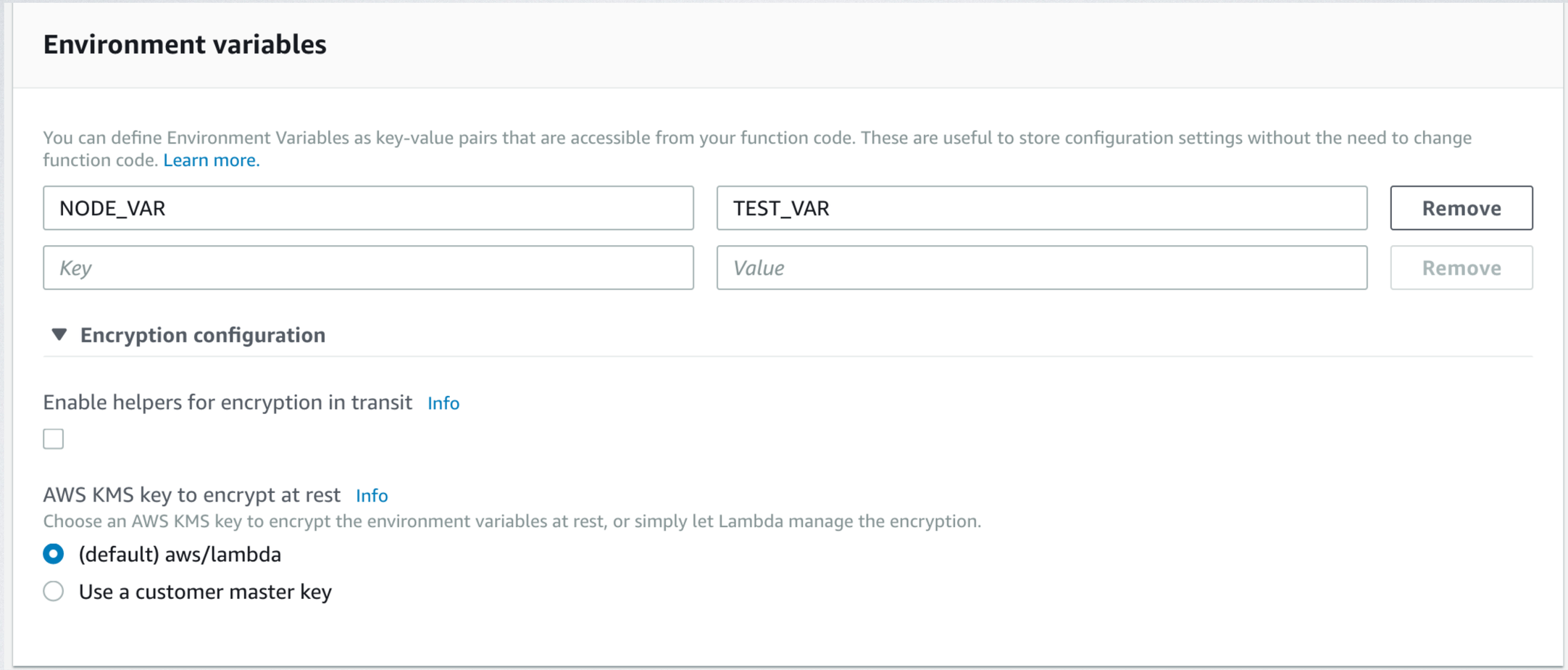
КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС

- Редактор кода

КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС

- Редактор кода
- Установка переменных окружения

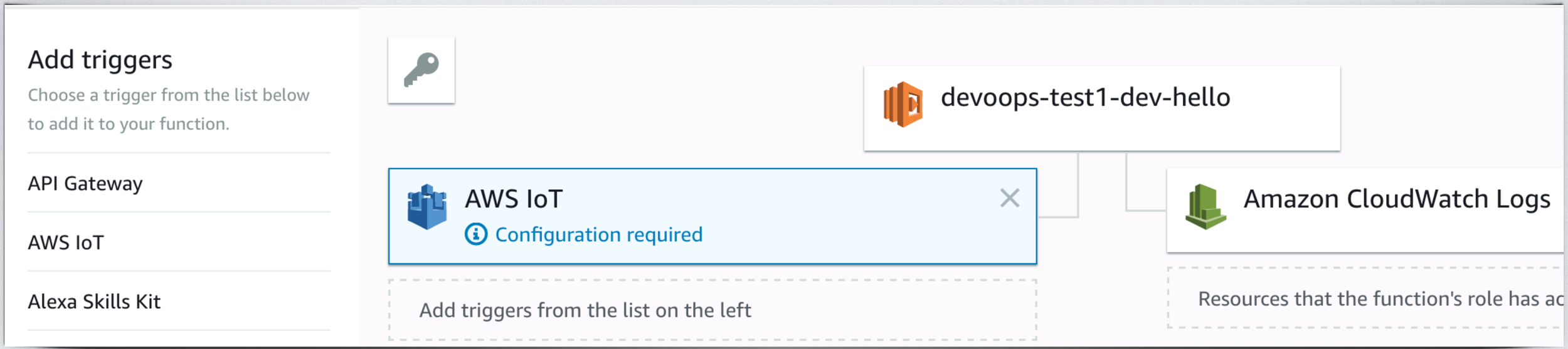
КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС



КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС

- Редактор кода
- Установка переменных окружения
- Удобно подписываться на события

КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС



КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС

- Нет скриптов для разворачивания окружения

КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС

- Нет скриптов для разворачивания окружения
- Трудности отладки функций

КОНФИГУРАЦИЯ ЧЕРЕЗ ИНТЕРФЕЙС

- Нет скриптов для разворачивания окружения
- Трудности отладки функций
- Проблемы в миграции

КОНФИГУРАЦИЯ

- Инфраструктура (Terraform, Ansible, CloudFormation)

КОНФИГУРАЦИЯ

- Инфраструктура (Terraform, Ansible, CloudFormation)
- AWS CLI

КОНФИГУРАЦИЯ

- Инфраструктура (Terraform, Ansible, CloudFormation)
- AWS CLI
  - Скрипты для разворачивания функций

КОНФИГУРАЦИЯ

- Инфраструктура (Terraform, Ansible, CloudFormation)
- AWS CLI
  - Скрипты для разворачивания функций
- Frameworks

КОНФИГУРАЦИЯ

- Инфраструктура (Terraform, Ansible, CloudFormation)
- AWS CLI
  - Скрипты для разворачивания функций
- Frameworks
  - Унифицированный yml конфиг

КОНФИГУРАЦИЯ

- Инфраструктура (Terraform, Ansible, CloudFormation)
- AWS CLI
  - Скрипты для разворачивания функций
- Frameworks
  - Унифицированный yml конфиг
  - Автоматическое создание API Gateway, SQS и тд

FRAMEWORKS

- Apex (http://apex.run/)

FRAMEWORKS

- Apex (http://apex.run/)
- Zappa (https://www.zappa.io/)

FRAMEWORKS

- Apex (http://apex.run/)
- Zappa (https://www.zappa.io/)
- Sparta (http://gosparta.io/)

FRAMEWORKS

- Apex (http://apex.run/)
- Zappa (https://www.zappa.io/)
- Sparta (http://gosparta.io/)
- Serverless (https://serverless.com/)

SERVERLESS FRAMEWORK

- JavaScript

SERVERLESS FRAMEWORK

- JavaScript
- Plugins

SERVERLESS FRAMEWORK

- JavaScript
- Plugins
- Абстракция над облаками

SERVERLESS FRAMEWORK - PLUGINS

- Запуск функции локально

SERVERLESS FRAMEWORK - PLUGINS

- Запуск функции локально
- Эмуляция ApiGateway, SQS и так далее

SERVERLESS FRAMEWORK - PLUGINS

- Запуск функции локально
- Эмуляция ApiGateway, SQS и так далее
- Логирование и мониторинг

КОНФИГУРАЦИЯ

```
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Creating Stack...
Serverless: Checking Stack create progress...
CloudFormation - CREATE_IN_PROGRESS - AWS::CloudFormation::Stack - devoops-test1-dev
CloudFormation - CREATE_IN_PROGRESS - AWS::S3::Bucket - ServerlessDeploymentBucket
CloudFormation - CREATE_IN_PROGRESS - AWS::S3::Bucket - ServerlessDeploymentBucket
CloudFormation - CREATE_COMPLETE - AWS::S3::Bucket - ServerlessDeploymentBucket
CloudFormation - CREATE_COMPLETE - AWS::CloudFormation::Stack - devoops-test1-dev
Serverless: Stack create finished...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service .zip file to S3 (576 B)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
CloudFormation - UPDATE_IN_PROGRESS - AWS::CloudFormation::Stack - devoops-test1-dev
CloudFormation - CREATE_IN_PROGRESS - AWS::IAM::Role - IamRoleLambdaExecution
CloudFormation - CREATE_IN_PROGRESS - AWS::IAM::Role - IamRoleLambdaExecution
CloudFormation - CREATE_IN_PROGRESS - AWS::Logs::LogGroup - HelloLogGroup
CloudFormation - CREATE_IN_PROGRESS - AWS::Logs::LogGroup - HelloLogGroup
CloudFormation - CREATE_COMPLETE - AWS::Logs::LogGroup - HelloLogGroup
CloudFormation - CREATE_COMPLETE - AWS::IAM::Role - IamRoleLambdaExecution
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Function - HelloLambdaFunction
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Function - HelloLambdaFunction
CloudFormation - CREATE_COMPLETE - AWS::Lambda::Function - HelloLambdaFunction
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Version - HelloLambdaVersionWPuMBrWEiE5lUkZJv3ujxFHBFVx7GyQtV3RXWmFi4o
CloudFormation - CREATE_IN_PROGRESS - AWS::Lambda::Version - HelloLambdaVersionWPuMBrWEiE5lUkZJv3ujxFHBFVx7GyQtV3RXWmFi4o
CloudFormation - CREATE_COMPLETE - AWS::Lambda::Version - HelloLambdaVersionWPuMBrWEiE5lUkZJv3ujxFHBFVx7GyQtV3RXWmFi4o
CloudFormation - UPDATE_COMPLETE_CLEANUP_IN_PROGRESS - AWS::CloudFormation::Stack - devoops-test1-dev
CloudFormation - UPDATE_COMPLETE - AWS::CloudFormation::Stack - devoops-test1-dev
Serverless: Stack update finished...
```

CONTINUOUS DELIVERY

- Внешний сервис (CodeShip)

CONTINUOUS DELIVERY

- Внешний сервис (CodeShip)
  - Сборка zip архива в докере

CONTINUOUS DELIVERY

- Внешний сервис (CodeShip)
  - Сборка zip архива в докере
  - Загрузка на S3

CONTINUOUS DELIVERY

- Внешний сервис (CodeShip)
  - Сборка zip архива в докере
  - Загрузка на S3
  - Автоматическое переключение в девелопе

CONTINUOUS DELIVERY

- Внешний сервис (CodeShip)
  - Сборка zip архива в докере
  - Загрузка на S3
  - Автоматическое переключение в девелопе
  - Ручное переключение в продакшене

CONTINUOUS DELIVERY

- Для одного региона: внешний CD

CONTINUOUS DELIVERY

- Для одного региона: внешний CD
- Для мультирегиона: Jenkins, Teamcity

DATABASE

DATABASES

- DynamoDB

DATABASES

- DynamoDB
- SQL

SQL

При росте трафика мы получем ошибку при подключении к базе данных

SQL

При росте трафика мы получем ошибку при подключении к базе данных

SQLSTATE[08004] [1040] Too many connections

DATABASES

- Масштабируемость

DATABASES

- Масштабируемость
  - Горячие контейнеры

DATABASES

GET /users => getUsers()

PUT /user/:id => updateUser(id)

POST /user => saveUser()

DELETE /user/:id => deleteUser(id)

DATABASES

GET /users => getUsers()	30/30
PUT /user/:id => updateUser(id)	30/30
POST /user => saveUser()	30/30
DELETE /user/:id => deleteUser(id)	30/30

Итого: 120

DATABASES

GET /users => getUsers()	25/30
PUT /user/:id => updateUser(id)	10/30
POST /user => saveUser()	50/50
DELETE /user/:id => deleteUser(id)	20/30

Итого: 140

DATABASES

SQLSTATE[08004] [1040] Too many connections

DATABASES

- Масштабируемость
  - Горячие контейнеры
  - Не все SQL решения могут работать с serverless

DATABASES

- Не использовать SQL

DATABASES

- Не использовать SQL
- Открывать и закрывать подключение

DATABASES

- Не использовать SQL
- Открывать и закрывать подключение
- Amazon Aurora Serverless

DATABASES

- Не использовать SQL
- Открывать и закрывать подключение
- Amazon Aurora Serverless
- Не использовать Serverless

ОТКРЫТЫЕ ВОПРОСЫ

- Может нам нужен моно репозиторий?

ОТКРЫТЫЕ ВОПРОСЫ

- Может нам нужен моно репозиторий?
- Как поддерживать версионность Lambda?

ОТКРЫТЫЕ ВОПРОСЫ

- Может нам нужен моно репозиторий?
- Как поддерживать версионность Lambda?
- Нужно ли нам столько Lambda functions?

ЧТО В ИТОГЕ

- Быстрая разработка и проверка теорий

ЧТО В ИТОГЕ

- Быстрая разработка и проверка теорий
- Быстрое масштабирование

ЧТО В ИТОГЕ

- Быстрая разработка и проверка теорий
- Быстрое масштабирование
- Постепенно лучше объединять в сервисы

ЧТО В ИТОГЕ

- Быстрая разработка и проверка теорий
- Быстрое масштабирование
- Постепенно лучше объединять в сервисы
- Логирование в ELK / Внешний сервис

ЧТО В ИТОГЕ

- Быстрая разработка и проверка теорий
- Быстрое масштабирование
- Постепенно лучше объединять в сервисы
- Логирование в ELK / Внешний сервис
- Учитывать особенности внешних приложений

ЧТО В ИТОГЕ

- Быстрая разработка и проверка теорий
- Быстрое масштабирование
- Постепенно лучше объединять в сервисы
- Логирование в ELK / Внешний сервис
- Учитывать особенности внешних приложений
- Serverless framework

ВОПРОСЫ

СПАСИБО

Email: ruslan.ru@gmail.com Twitter: @ruslan_firefly